Machine Learning Foundations

(a refresher)

William Becker

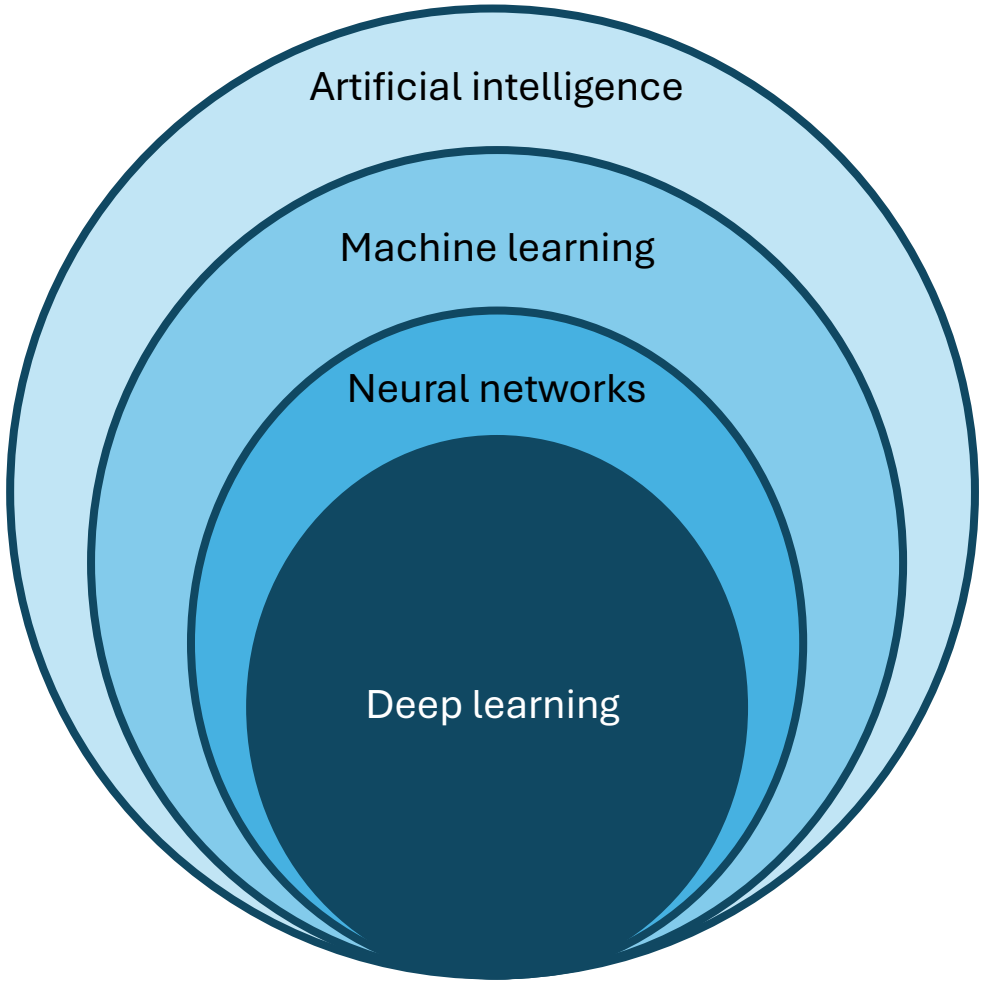
Based on slides from Sara Hahner and Jesper Dramsch

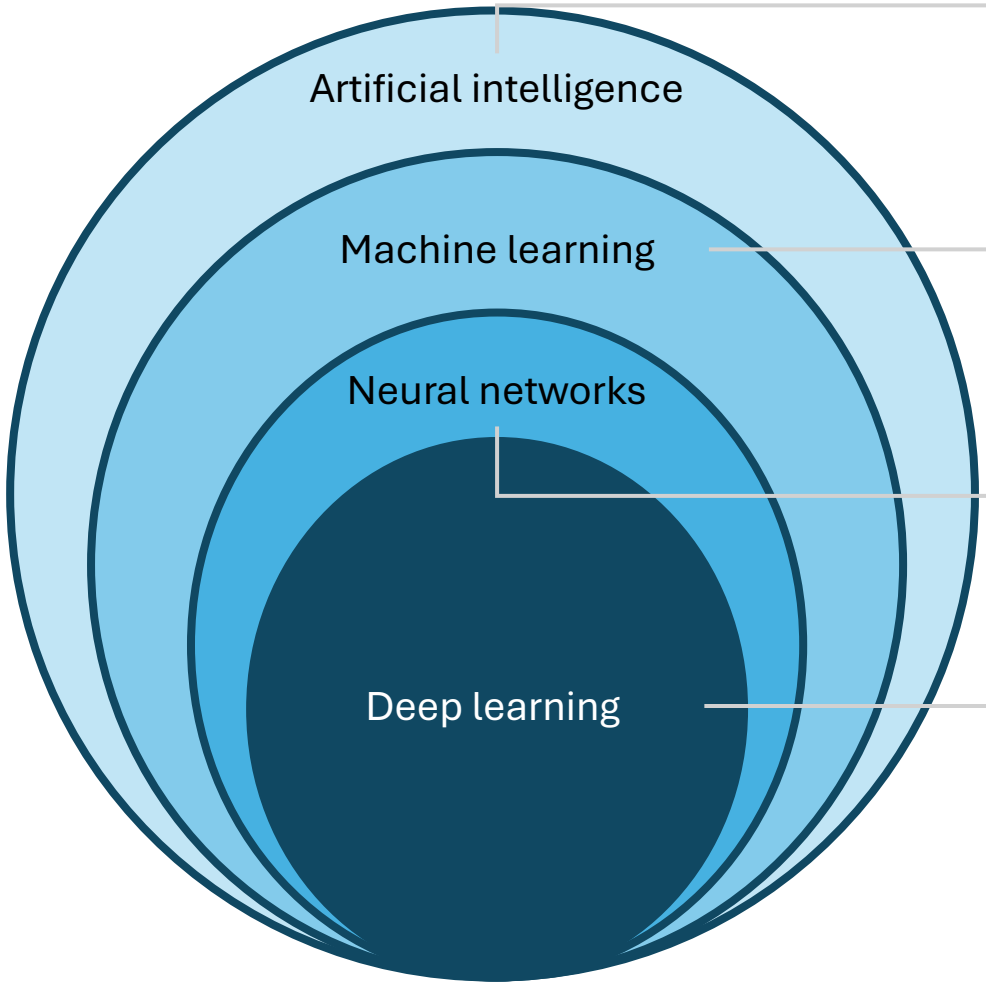
Outline

- Understanding AI & Machine Learning
- Types of Machine Learning
- Key Concepts in Machine Learning
- Dealing with Data
- Finding the Optimal Model

Understanding AI & ML

What’s the difference between machine learning and artificial intelligence?


Artificial intelligence
Machine learning
Neural networks
Deep learning



Performing tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making

Algorithms that can learn from data and generalise to unseen data, and thus perform tasks without explicit instructions.

Computational models inspired by the structure and functions of biological neural networks

The use of neural networks with multiple hidden layers of neurons (up to possibly thousands).

Machine learning

Algorithms that can learn from data and generalise to unseen data, and thus perform tasks without explicit instructions.

Google

Machine learning is a subset of artificial intelligence that enables a system to autonomously learn and improve using neural networks and deep learning, without being explicitly programmed, by feeding it large amounts of data.

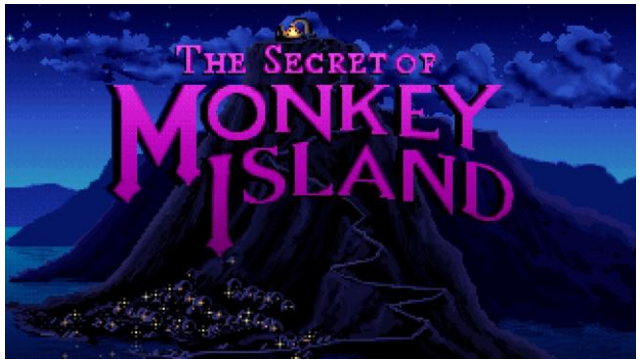
Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalise to unseen data, and thus perform tasks without explicit instructions.

IBM

Machine learning (ML) is a branch of artificial intelligence (AI) focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data.

ECMWF

Rule-based/symbolic AI



The Secret of Monkey Island, 1990

```
IF player_insult == "You fight like a dairy farmer"
AND response == correct_comeback
THEN player_wins_duel
```

```
IF player_has_item = rubber_chicken
AND location = cable
THEN allow puzzle solution
```

Expert systems were the prevailing way to encode knowledge into computer models/systems for many years.

Also robotics and control systems, etc.

ECMWF

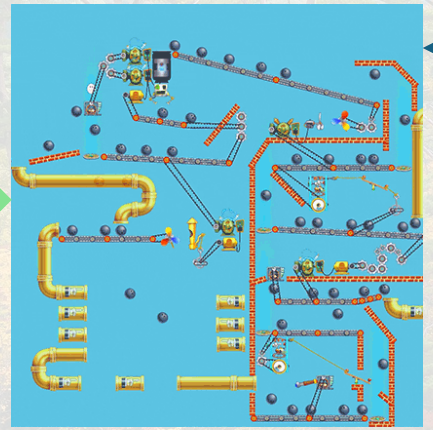
A tale of two modelling approaches

Observations of the world

Human learning

Machine learning

Initial/boundary conditions



Calibration, etc

RESULTS

Physics-based model, manually encoded based on our understanding of the world

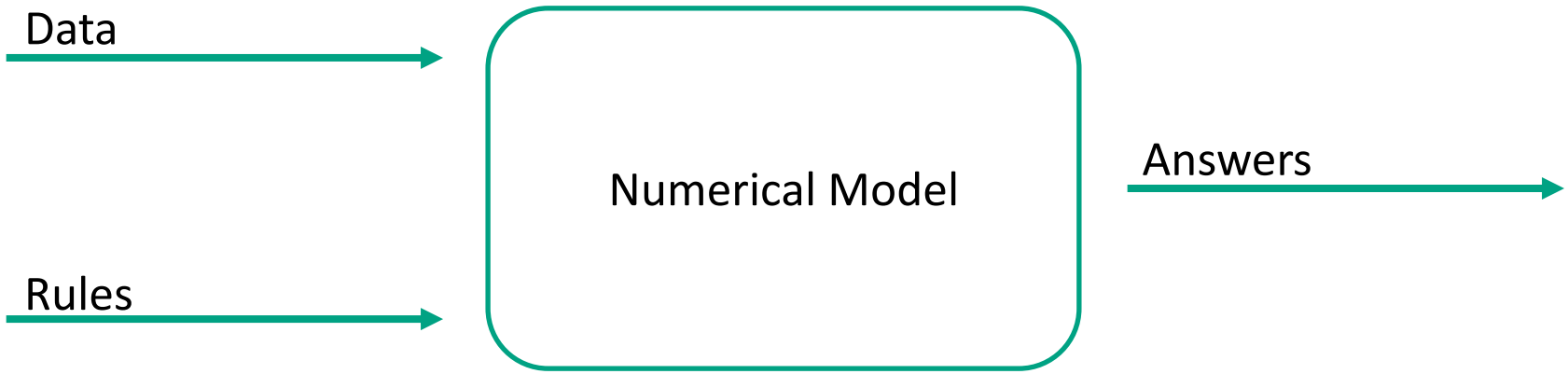
Initial/boundary conditions

RESULTS

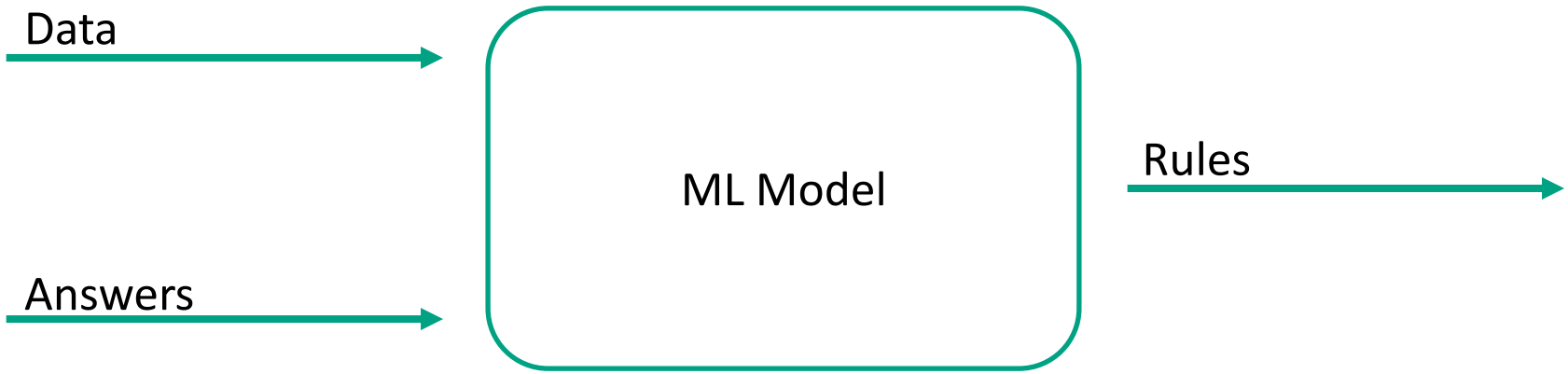
Data-driven model, learned “automatically” from a set of training data.

Black boxes?

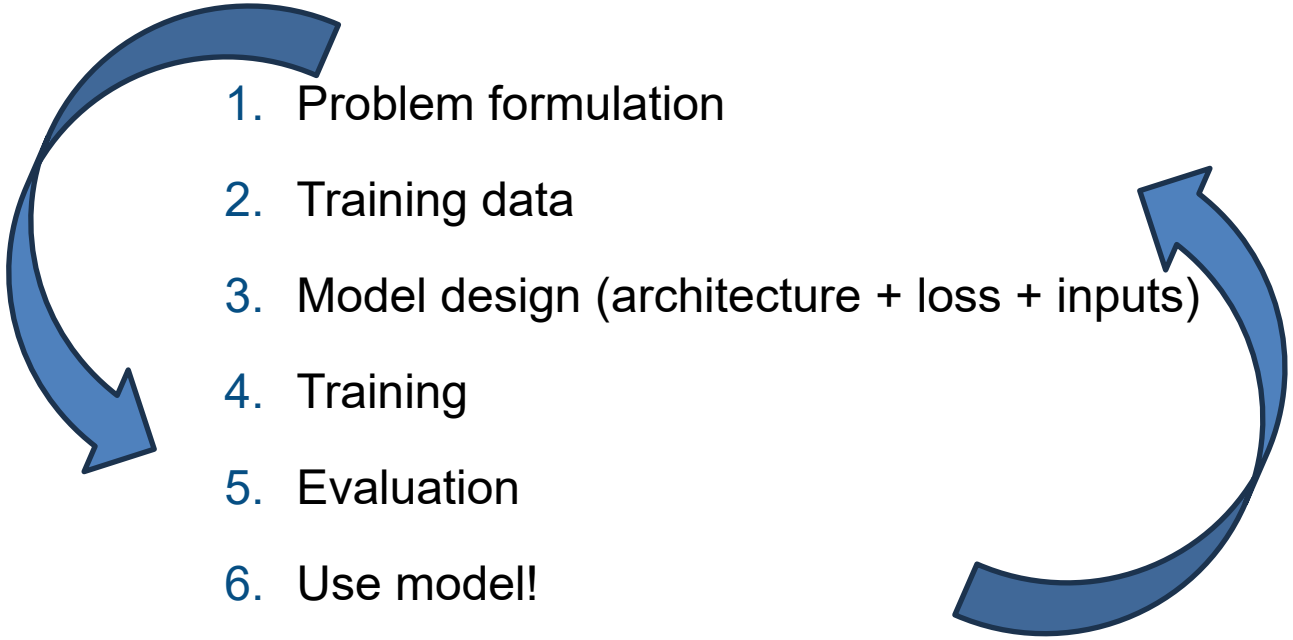
Classical Modelling



Supervised Machine Learning Modelling



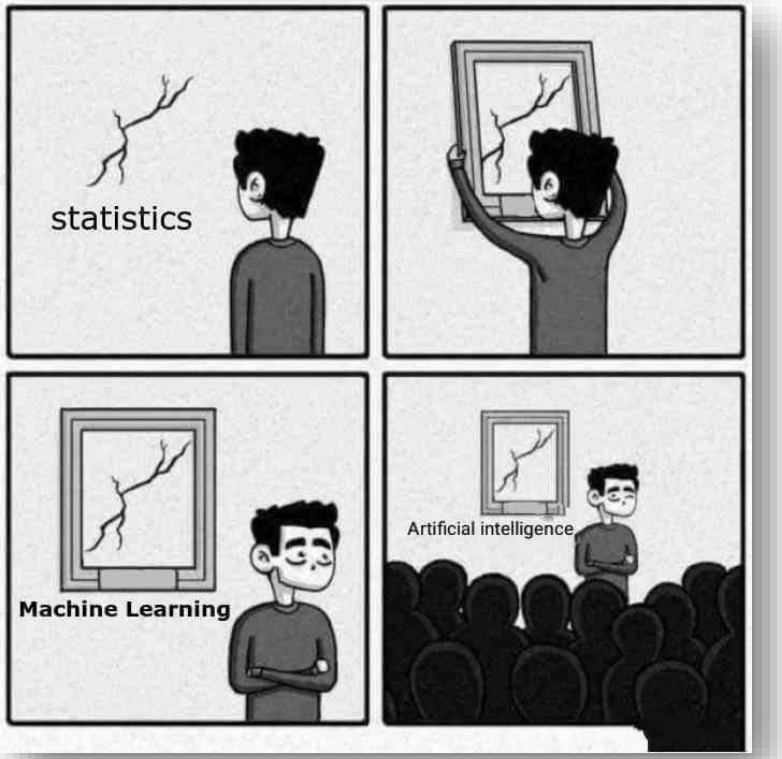
Very Simplified ML workflow



1. Problem formulation
2. Training data
3. Model design (architecture + loss + inputs)
4. Training
5. Evaluation
6. Use model!

These steps can be more or less complex, depending on the problem.

Stats in disguise?



ECMWF

Stats in disguise?

Machine learning is rooted in statistics (optimisation, probability, weights, etc). BUT has become a distinct field distinguished by:

- Very large data sets
- Capacity to deal with very high dimensional spaces – large models of billions of parameters, highly complex relationships
- Leveraging large computational resources, GPUs, computer science
- Ability to deal with unstructured, non-rectangular data (images, video, spatiotemporal, multimodal, etc.)
- Emphasis on empirical results rather than asymptotic/theoretical properties

It's all hype and marketing

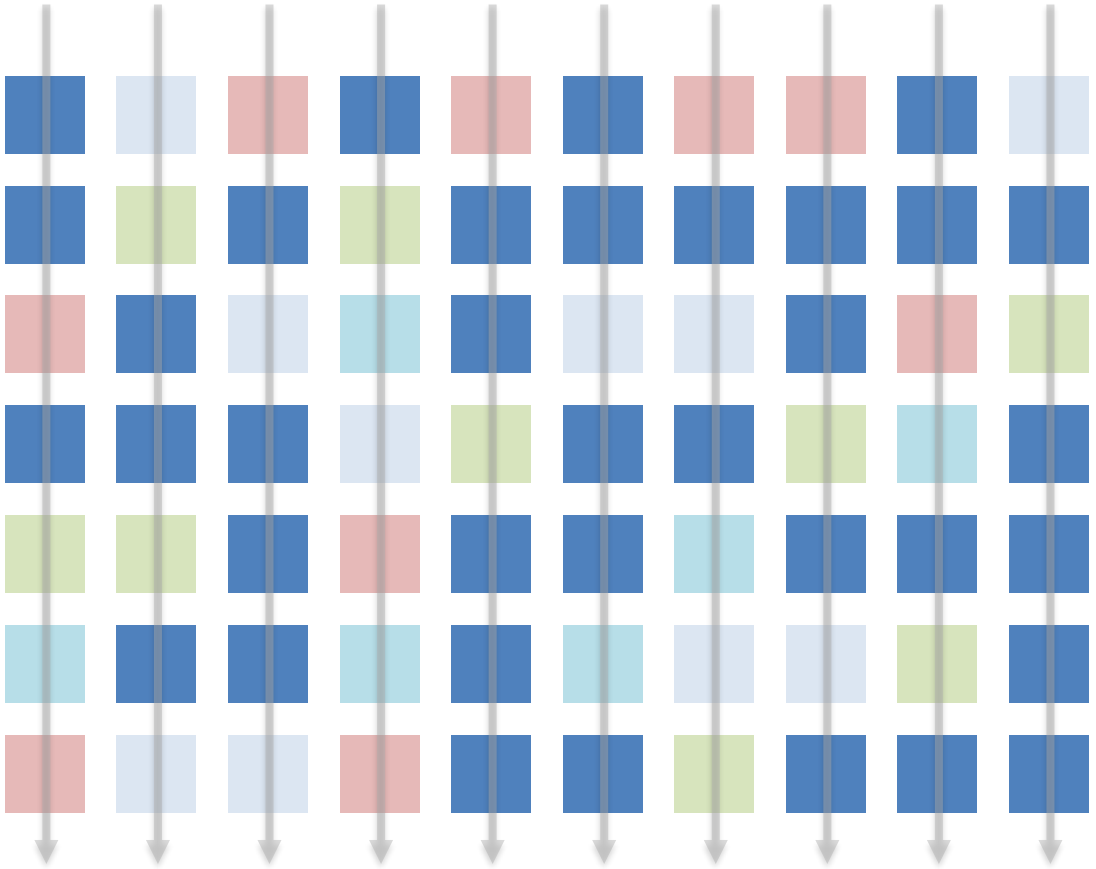
The truth (somewhere here, as usual)

AI can do anything!

Sceptics — Evangelists

Types of Machine Learning

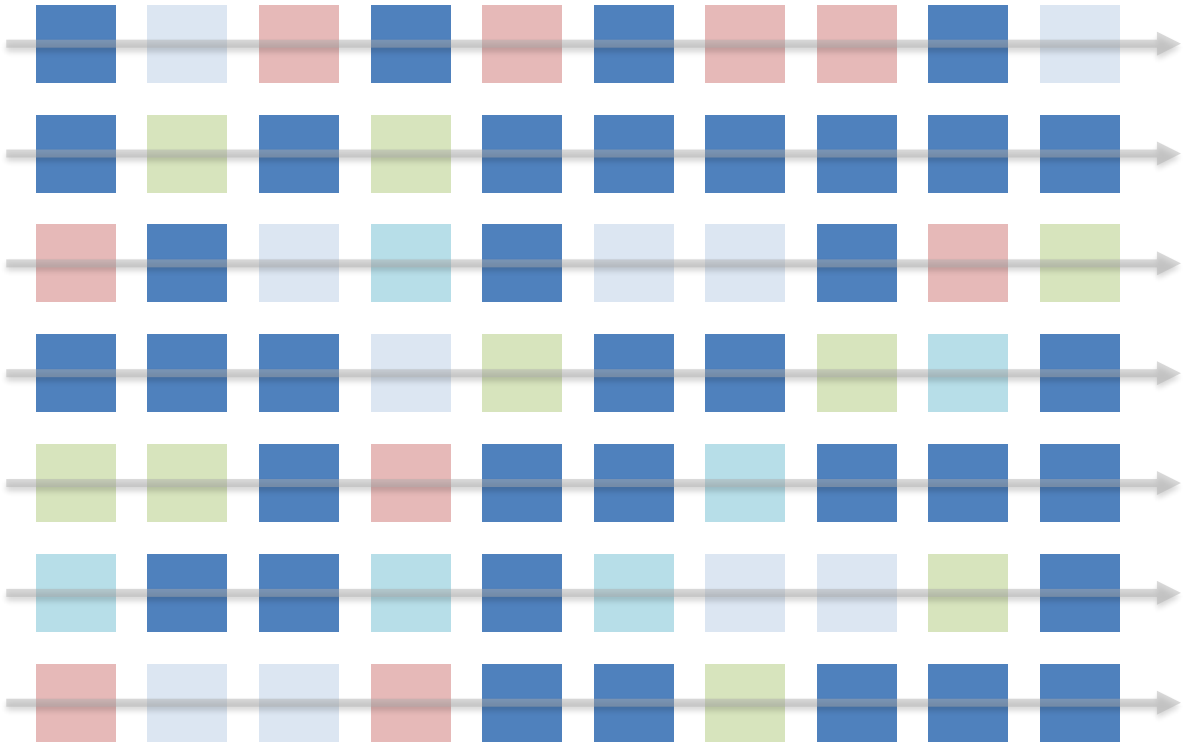
Features/variables



Pixel values, meteorological variables, tokens, etc.

ECMWF

Observations/samples



Weather at a certain time

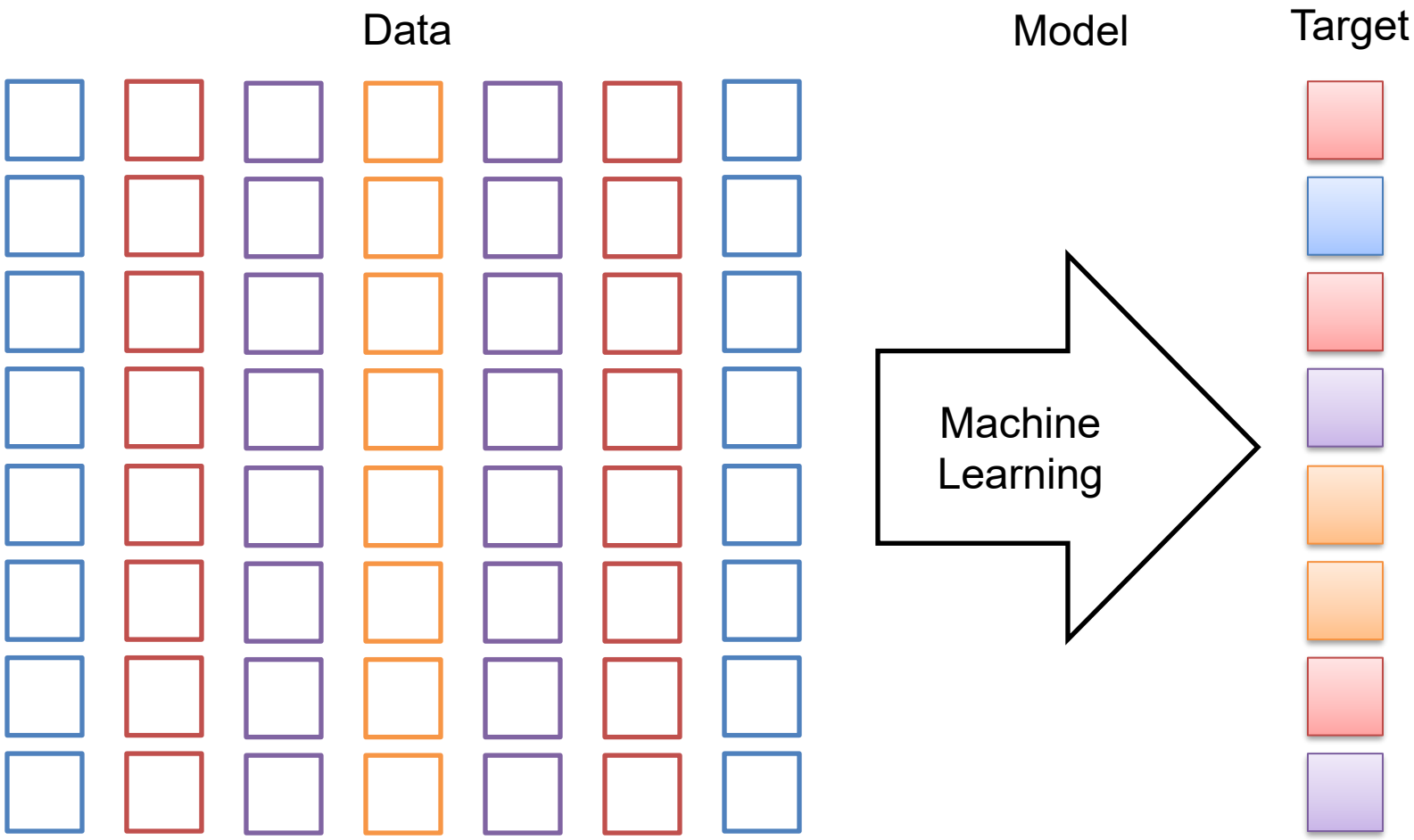
Weather at another time

Weather at another time

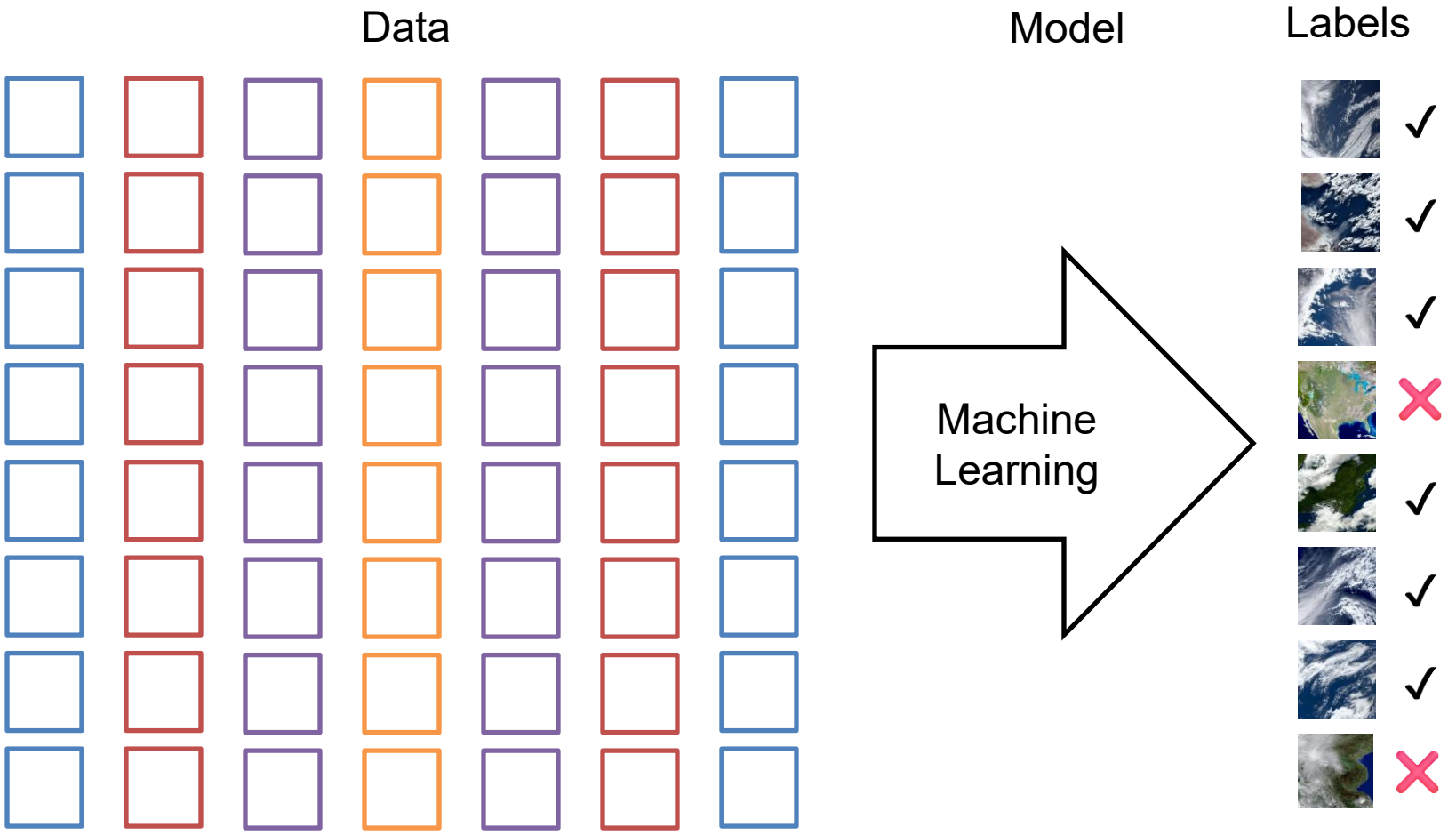
etc.

At different points in space, time or repeated random sampling, etc.

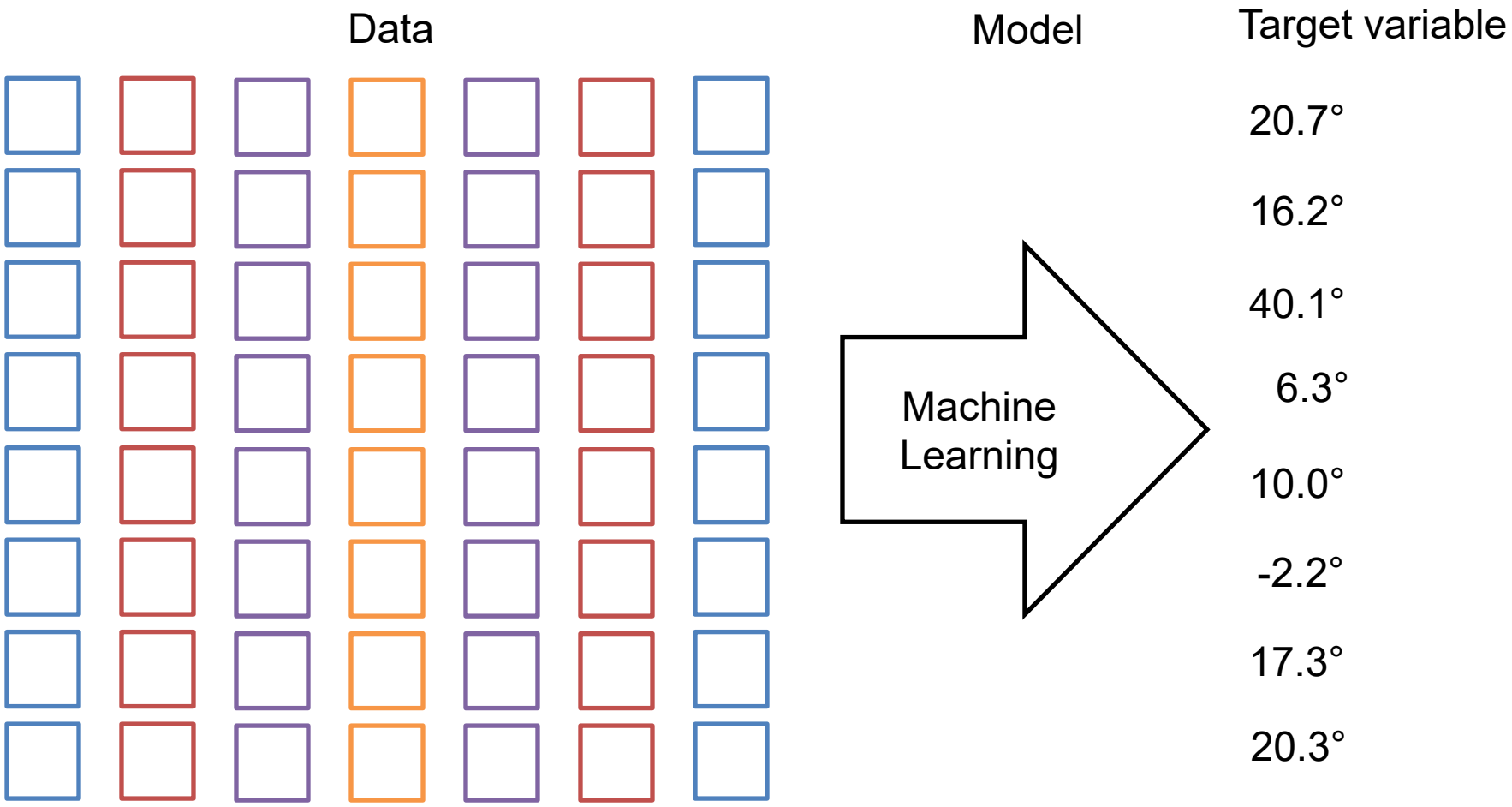
Supervised Learning



Supervised Learning – Classification

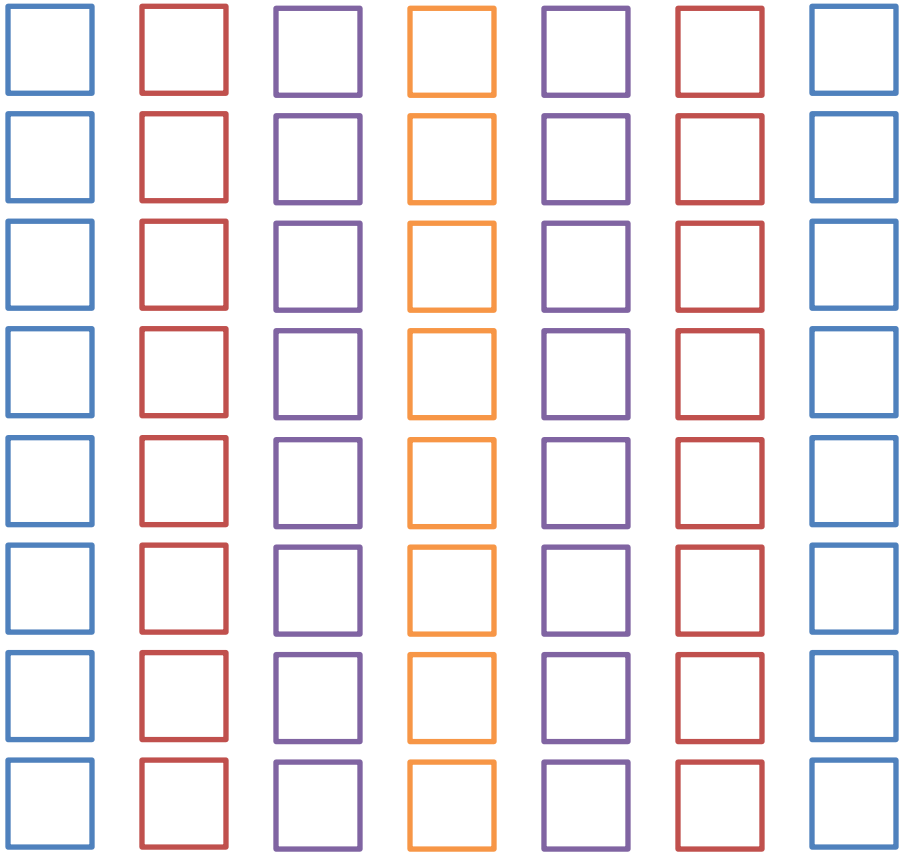


Supervised Learning – Regression



Unsupervised Learning

Data



Labels

?

?

?

?

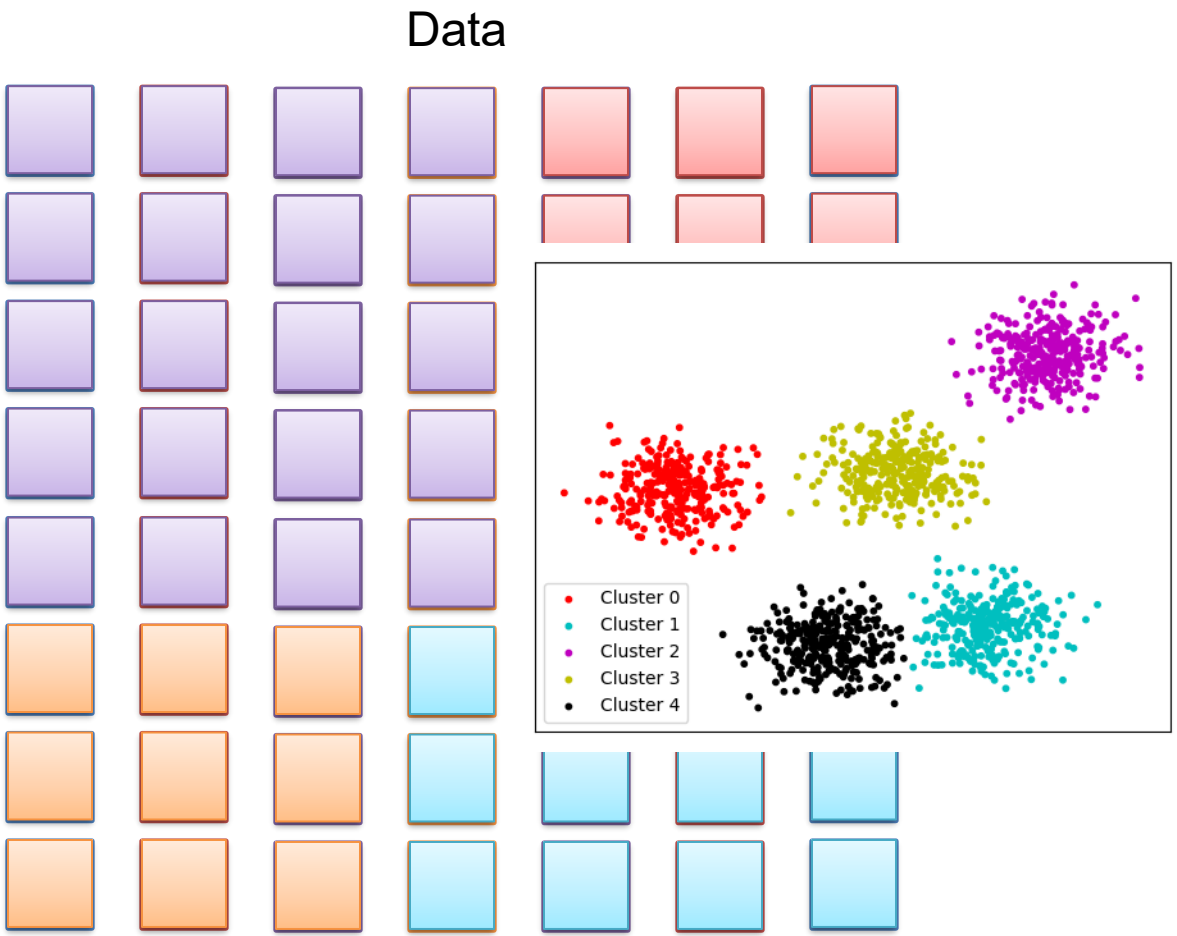
?

?

?

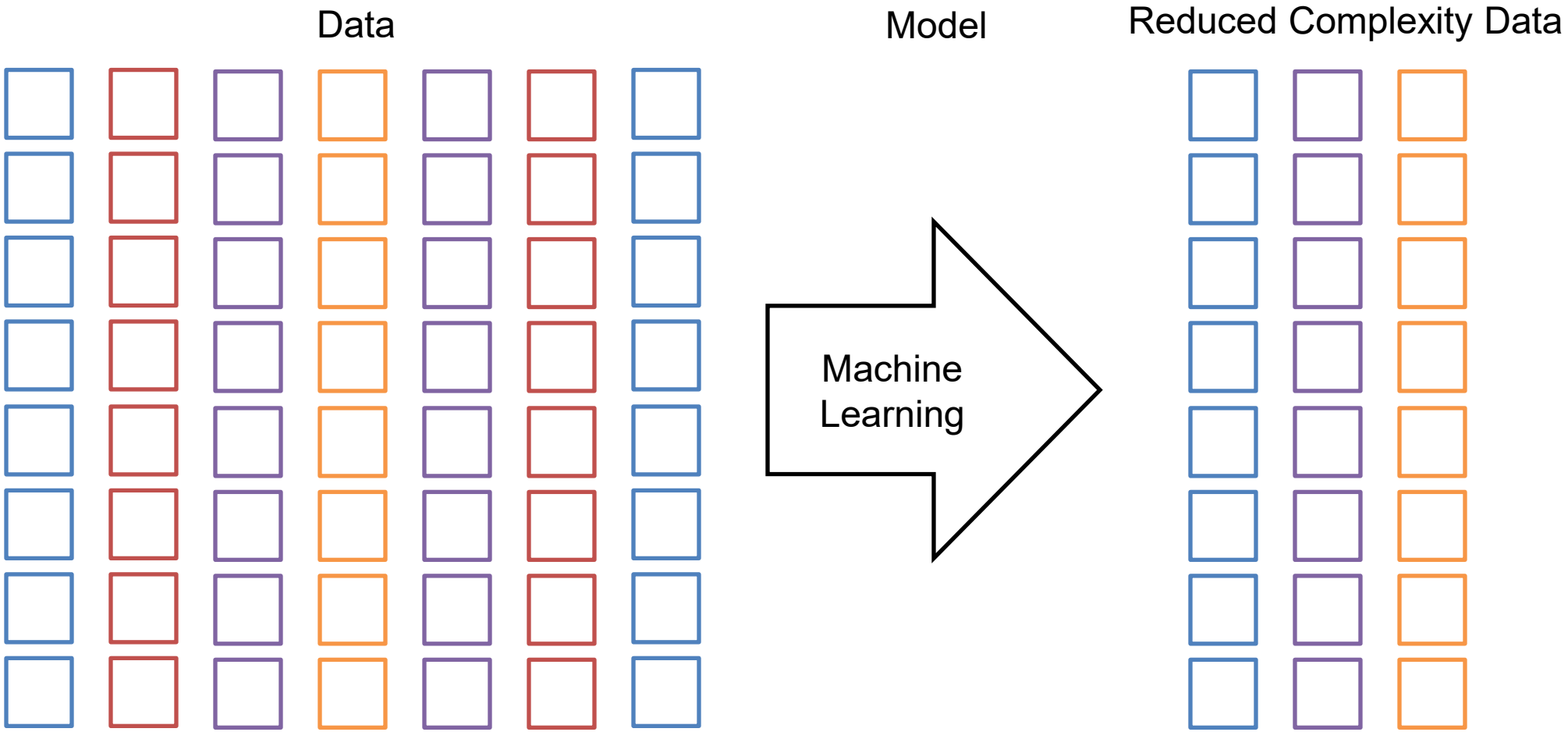
?

Unsupervised Learning – Clustering



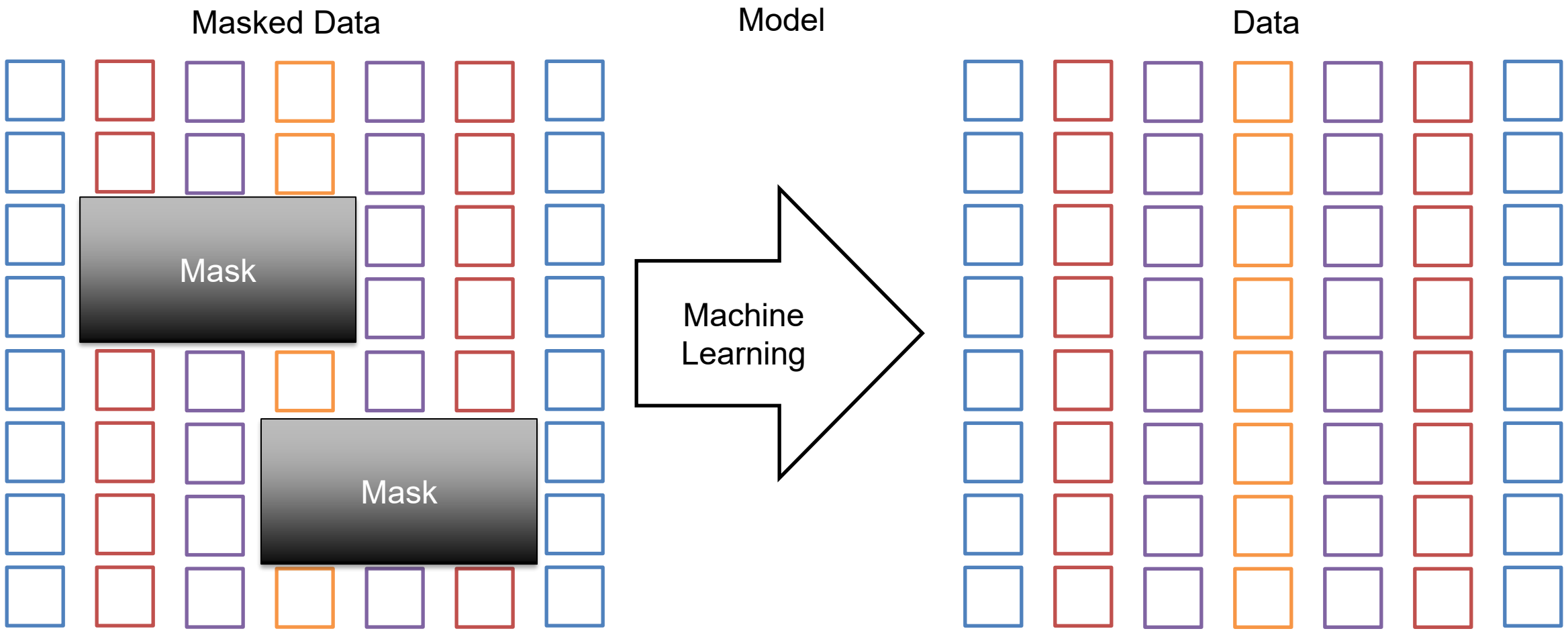
ECMWF

Unsupervised Learning – Dimensionality Reduction



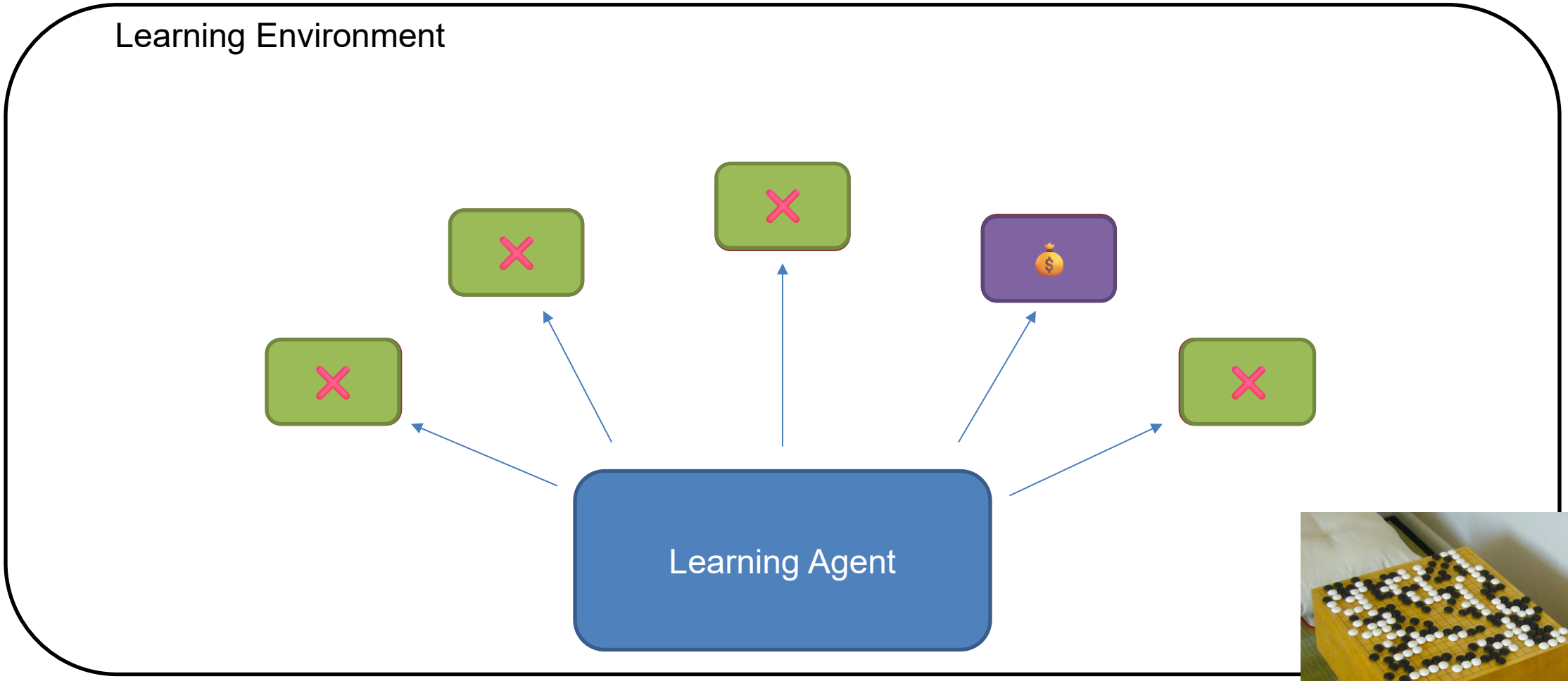
Self-supervised Learning

Wednesday at 11:45
Self-supervised Learning

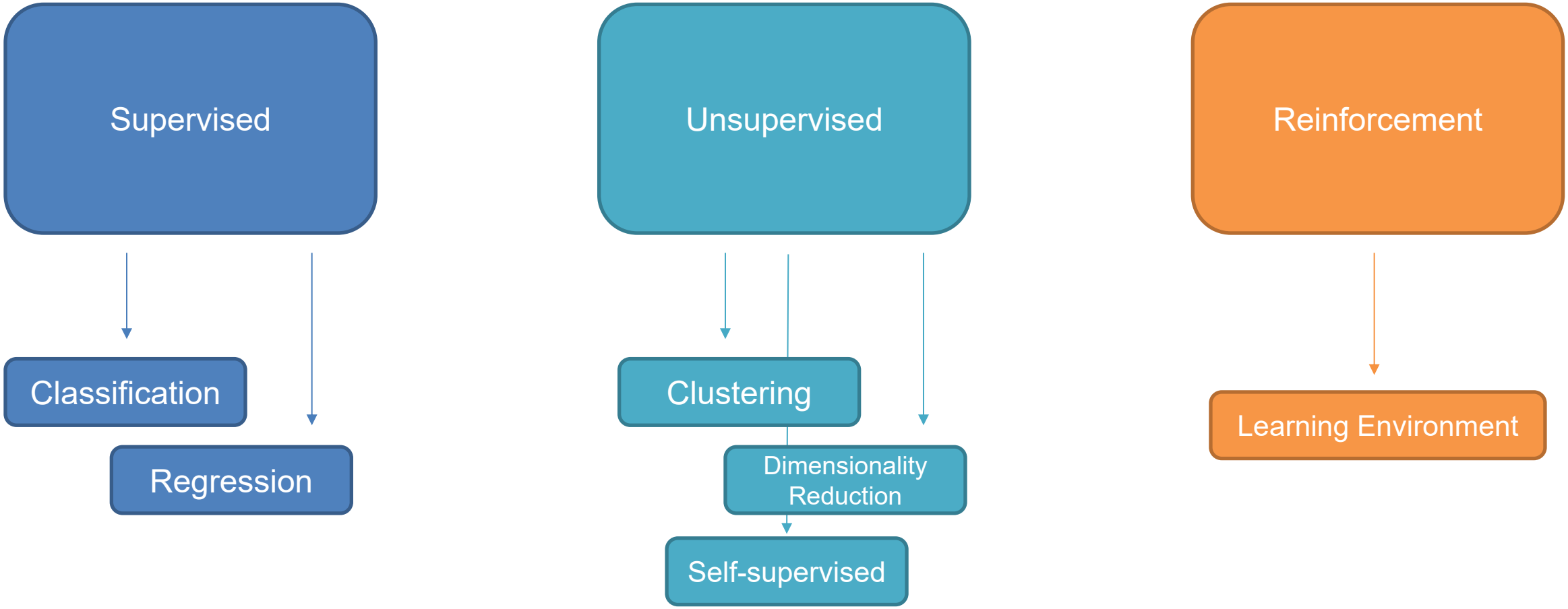


ECMWF

Reinforcement Learning



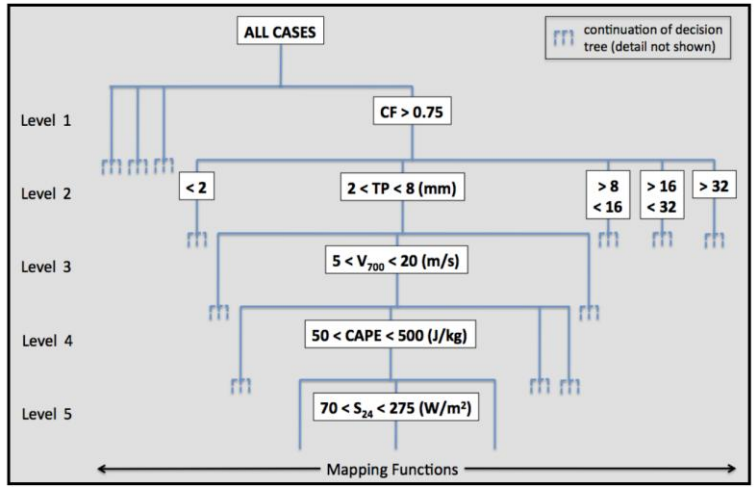
Types of Machine Learning



ECMWF

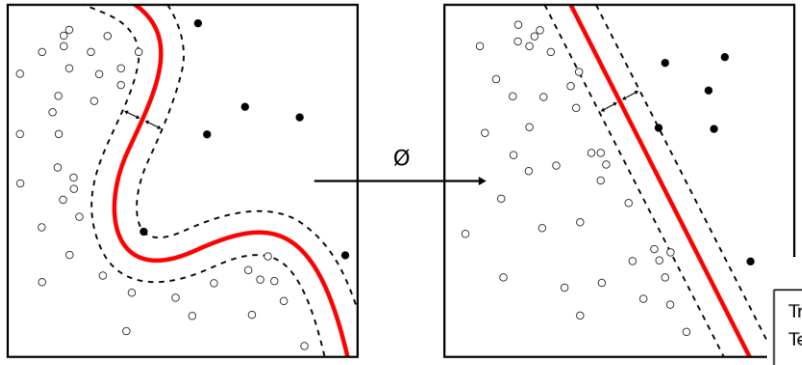
What models?

Decision Trees

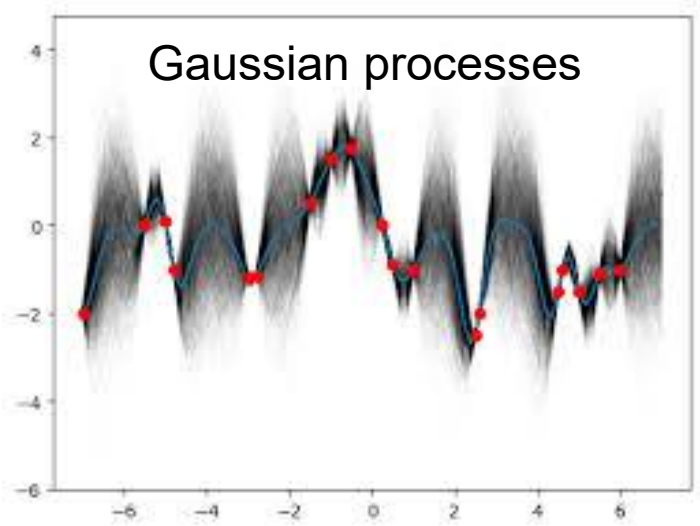


Hewson and Pillosu 2020

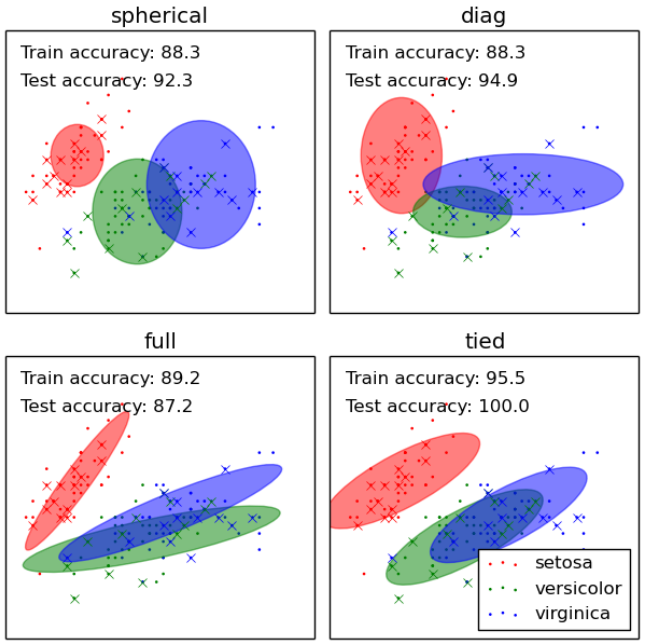
Support-Vector Machines



Gaussian processes

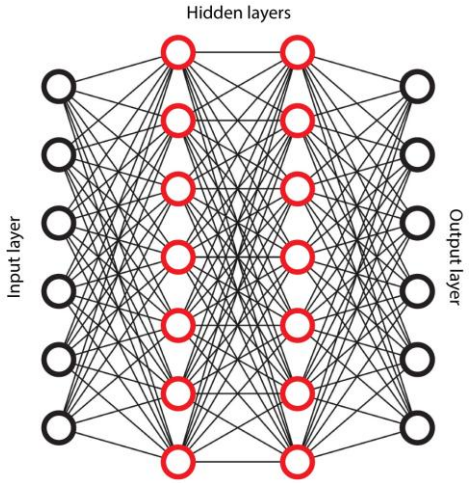


Mixture models



What models?

Neural Networks



15:30 Neural Networks
and Deep Learning

Common ML modelling is facilitated by open source software

Choose a model

```
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

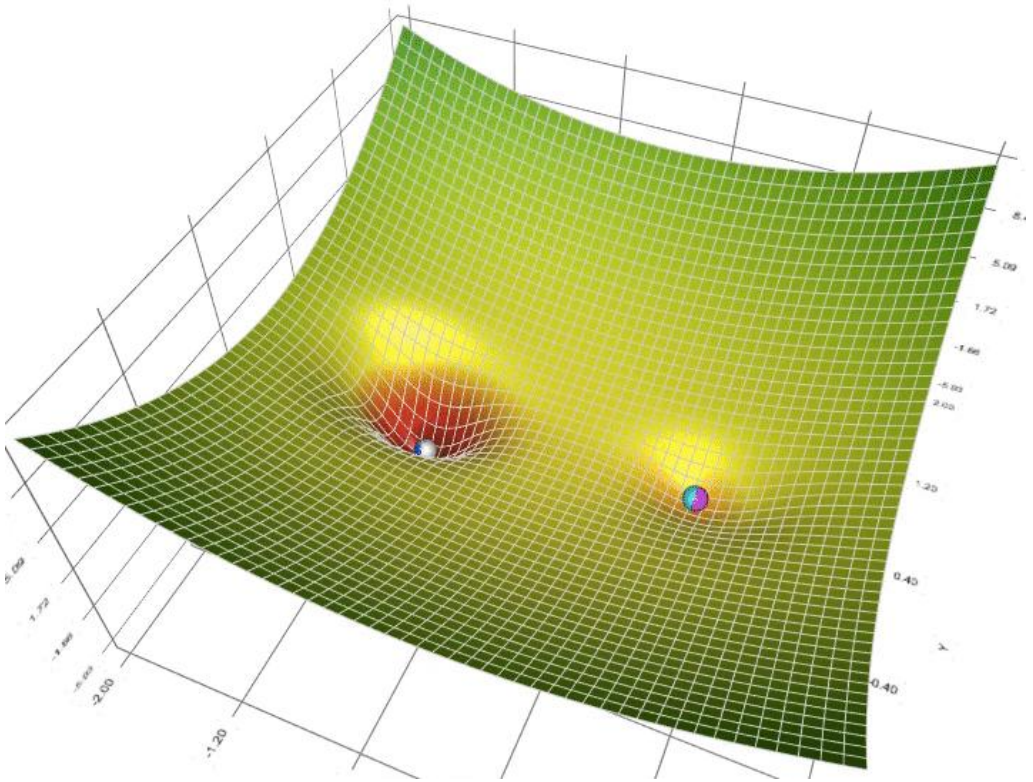
Fit the model to training data

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, gamma=0.001)
```

Use model to predict

```
>>> clf.predict(digits.data[-1:])
array([8])
```

The **Learning** in ML: Numerical Optimisation



All ML models have **parameters** which need to be trained, using the training data.

We want to minimise the difference between model outputs and observed training data (error).

- Model labels vs true labels
- Model regression values vs observed values
- Model forecast vs observed weather forecast

"Gradient Descent iteratively adjusts **model parameters** to minimise error, using the slope of the loss function to guide updates."

Key Concepts
Machine Learned Models

Training, Validation and Testing

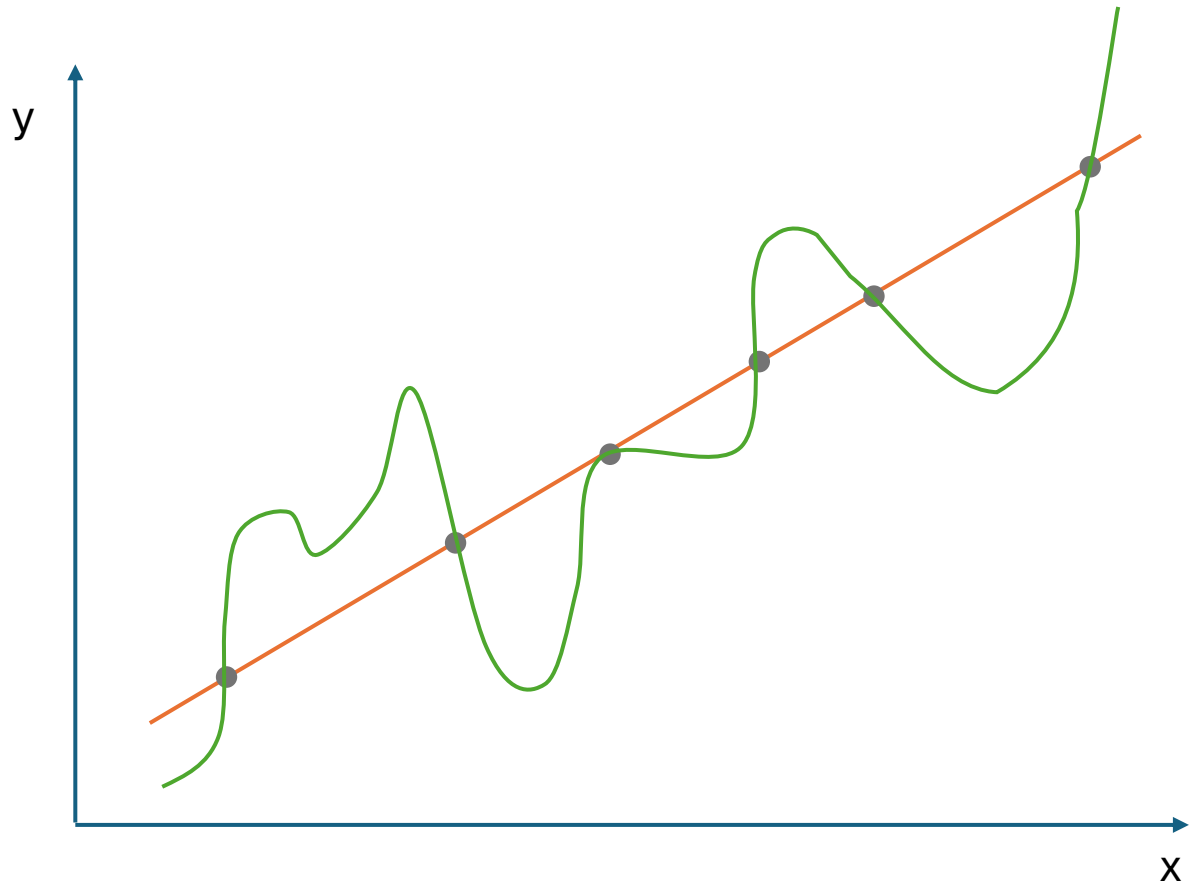
Disjoint Datasets	Purpose
Training Set	Fit the model (learn parameters)
Validation Set	Tune model architectures and select best model
Test Set	Final unbiased evaluation after training

Workflow

1. Train the model on training data
2. Validation using validation set -> adjust model 
3. Test the final model on test data -> estimate true performance

“Generalization is a ML **model’s ability** to generate accurate and reliable predictions on previously **unseen data**.”

Overfitting



What relationship does y have with x? Which kind of model would be the best fit here?

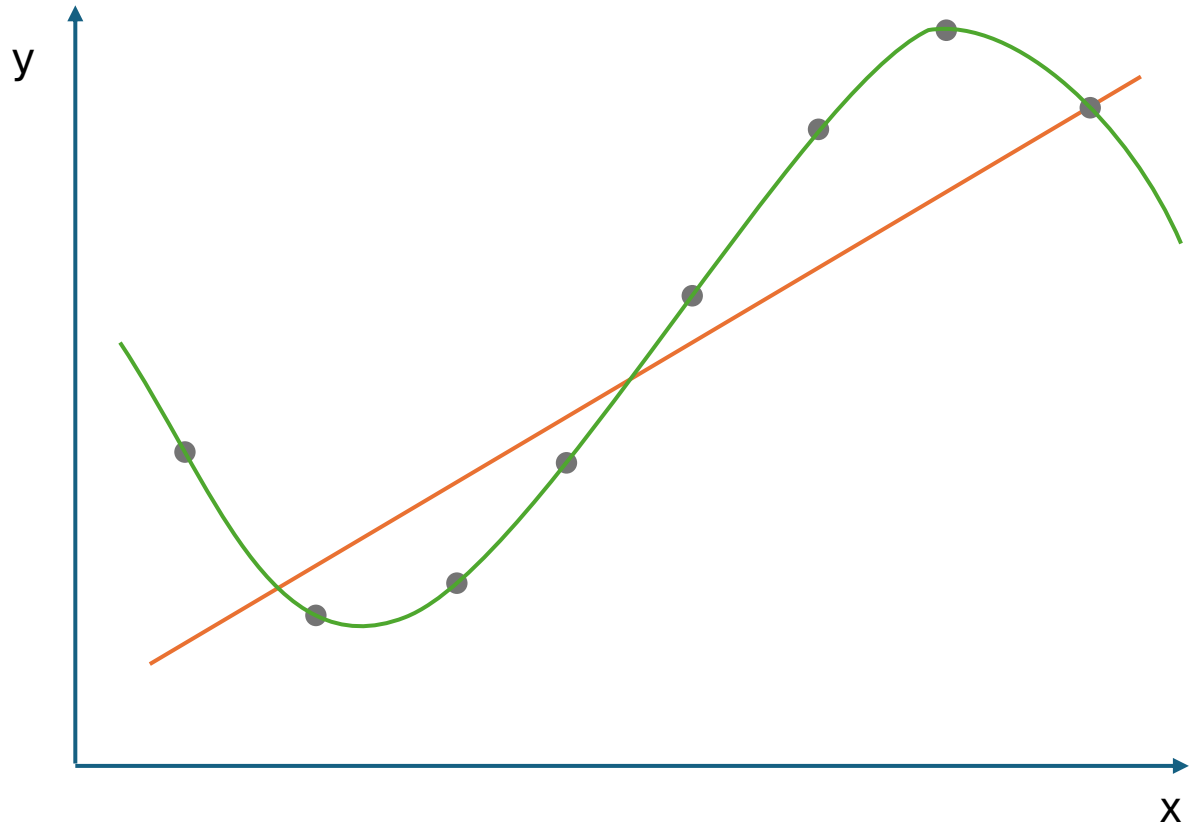
Another way to look at it – which model is more *likely* to produce a set of data with this pattern?

Occam's razor – states that the simplest model *that explains the data* is the most likely.

Overfitting is "remembering" the training data, but nothing else.

We want to generalize to unseen data.

Underfitting



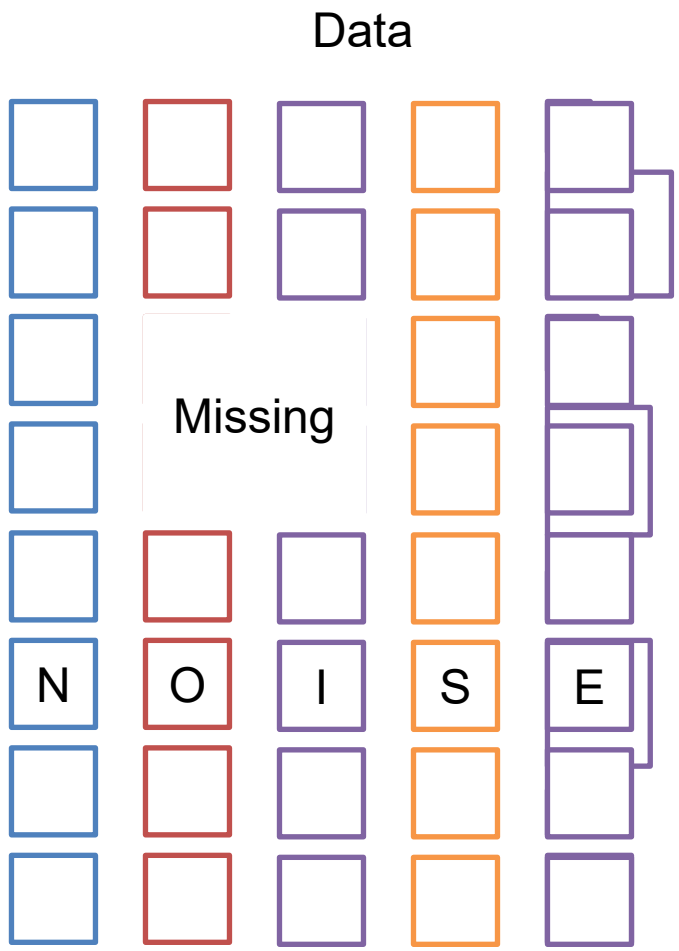
What relationship does y have with x? Which kind of model would be the best fit here?

Underfitting is when the model is not complex enough to explain the data.

In a way this is easier to spot and less of a problem in practice, since it would show up in your loss metrics, even on the training data set.

Dealing with Data

Data Preprocessing

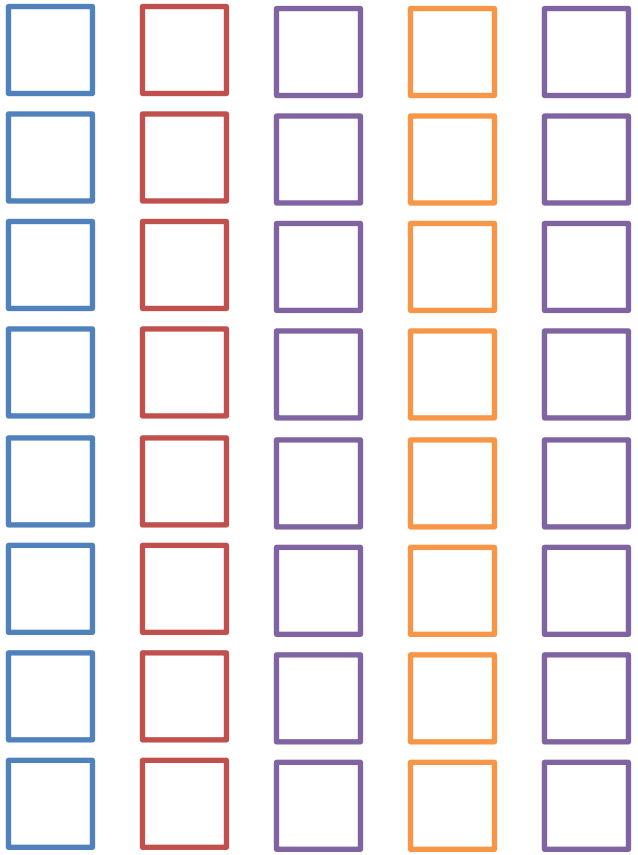


- Machine learning models struggle with irregular data
- Imputation
  - Filling in missing values
  - Often with Mean or Median
- Data Cleaning
  - Removing noise from data
  - Careful! Easy to ”over-clean”
  - Needs to be faithful to real-world data
- Normalisation
  - Standardization
  - Min-Max Scaling
- Transformations
  - Log-Scaling

Managing Big Data with Batch Processing

Tomorrow at 12:00
Data Handling and
Infrastructure

Big Data



Batch 1

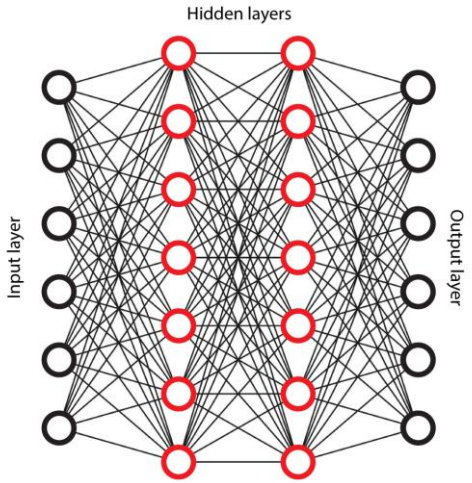
Batch 2

Hyperparameters and finding the optimal model

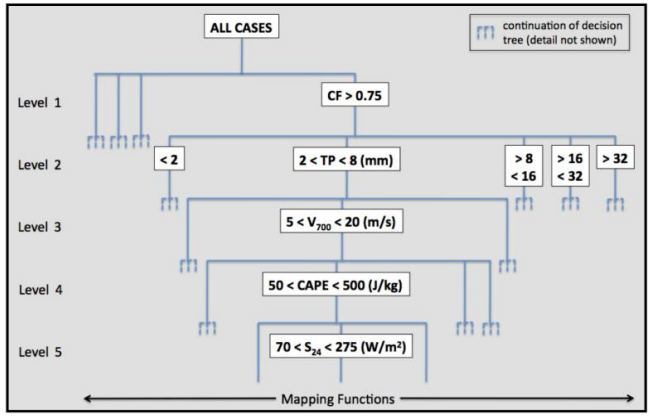
Hyperparameters and Tuning

- Parameters:
  - Learned from data during training
  - E.g. weights and biases in an ANN
- Hyperparameters:
  - ”Settings of Model” – YOU set these (mostly)
- Examples of Hyperparameters:
  - Number of nodes
  - Number of layers
  - Number of Trees
  - Learning Rate of optimization process
  - Batch size, of incremental training

Neural Networks

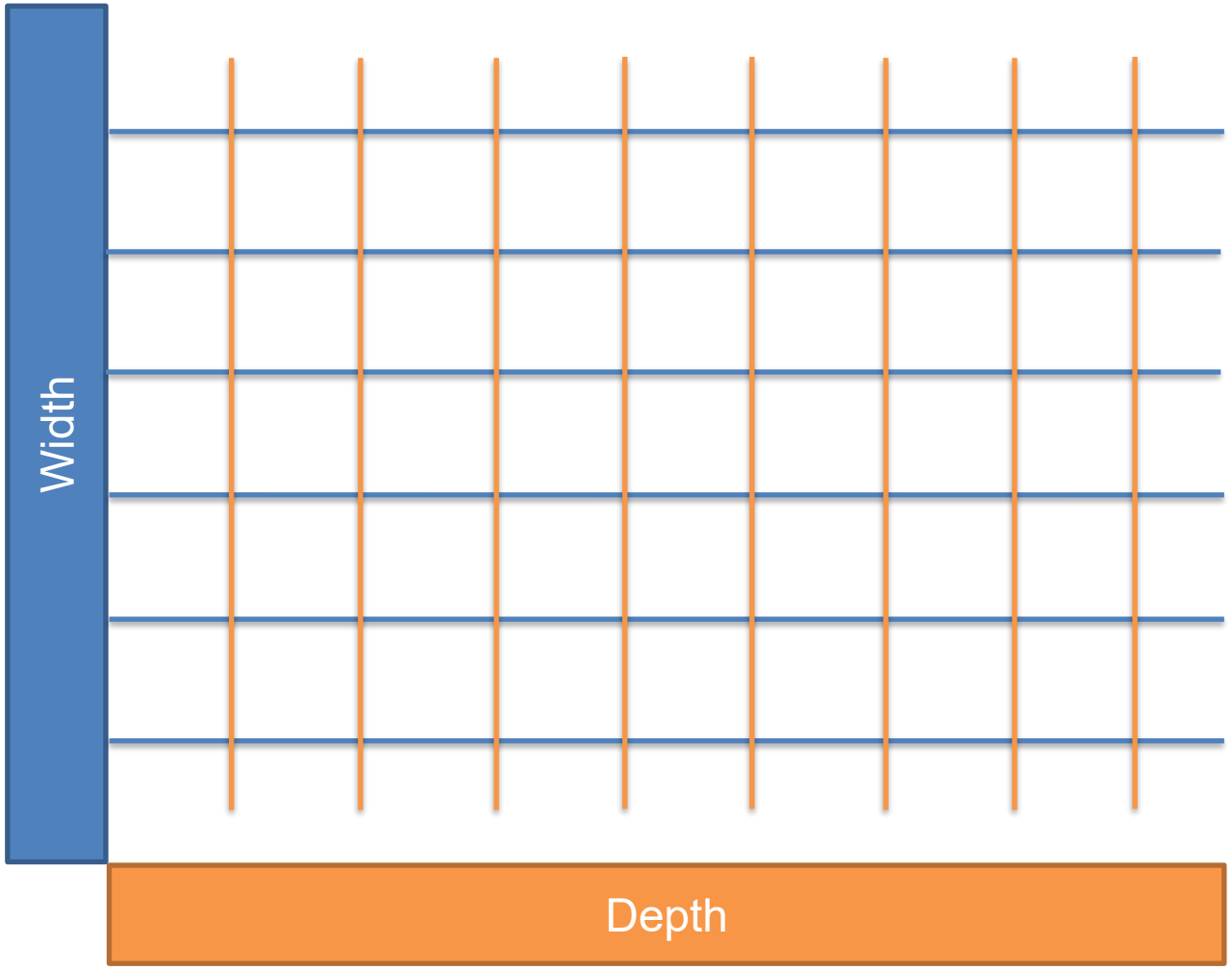


Decision Trees



Hewson and Pillosu 2020

Grid search

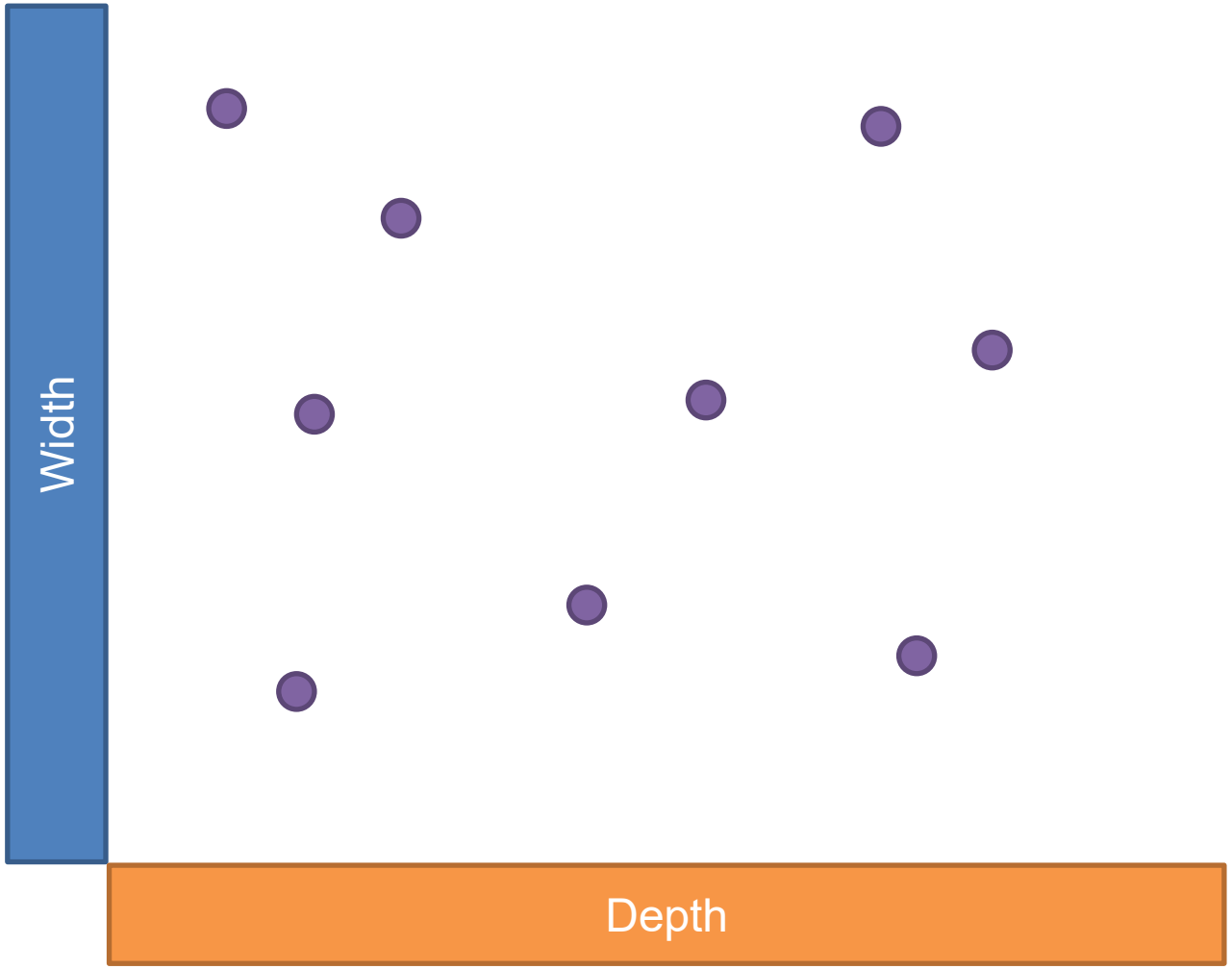


- Exhaustive Search
- Every Combination is Evaluated
- Combinatoric Explosion of Evals
- Inefficient searching beyond minimum
- Possible to miss optimal parameters

because explicit values are provided

```
1 from sklearn.model_selection import GridSearchCV
2
3 parameters = {'width':[5, 10, 15, 20],
4               'depth':[1, 2, 3, 4, 5, 6],
5               'activation':['tanh', 'relu']}
6
7 gridcv = GridSearchCV(neural_network, parameters)
8
9 gridcv.fit(X_train, y_train)
```

Randomized search



- Exhaustive Search
- Budget independent of No. parameters
- Adding Parameters not Inefficient
- Inefficient searching beyond minimum
- Possible to miss optimal parameters because explicit values are provided

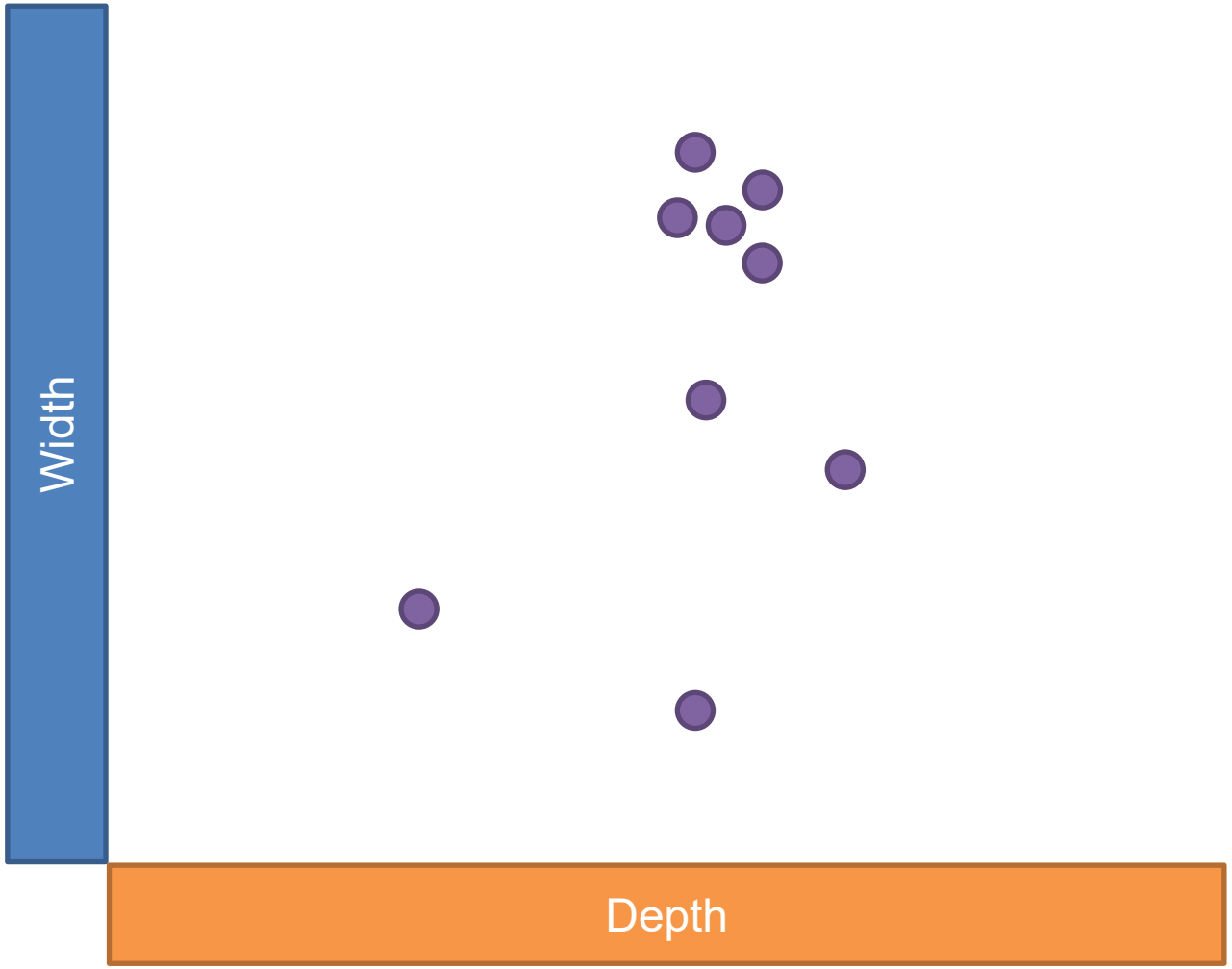
```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform

distributions = {'width': uniform(5, 15),
                 'depth': uniform(1, 5),
                'activation':['tanh', 'relu']}

randomcv = RandomizedSearchCV(neural_network, distributions)

randomcv.fit(X_train, y_train)
```

Bayesian search



- Search based on former parameters
- Bayesian Optimization
- Converges to a minimum
- Adding Parameters adds complexity
- Unimportant parameters complicate optimization significantly

```
from skopt import BayesSearchCV

distributions = {'width': (5, 20, 'uniform'),
                 'depth': (1, 6, 'uniform'),
                 'activation':['tanh', 'relu']}

randomcv = BayesSearchCV(neural_network, distributions)

randomcv.fit(X_train, y_train)
```

Final thoughts

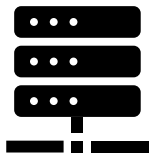
Nice problems

What makes a problem suitable for a data driven (ML) model?

Clearly defined objective

Framed in terms of inputs and outputs

- Predict the weather tomorrow based on the weather today
- Classify an image into one of 10 possibilities
- Generate text based on a prompt



Training data

Plentiful, good quality, informative, covers model use cases. Generally, the more the better (not for all ML methods though).

- ERA5 historical global weather data set
- ImageNet set of labelled images
- Text, articles, web pages from the internet

Modelling logic

Outputs can plausibly be inferred from inputs (domain knowledge). Simpler methods not appropriate.

- Tomorrow's weather *can* be inferred from today's (domain knowledge)
- Complex system justifies deep learning approach

But you still need

ML knowledge – architecture, algorithms, training

Compute – GPUs

Open-source software – PyTorch, Anemoi, etc.

Data infrastructure and engineering

Domain knowledge – understanding the problem

anemoi inference

What we Learned

- AI and Machine Learning are related but distinct
- Open-source software makes ML easier
- Types of machine learning model:
  - Un- and Supervised learning
  - Reinforcement Learning
- Other relevant “Learning”
  - Deep Learning
  - Transfer Learning
- Generalization and Overfitting
- Data-Preprocessing
- Hyperparameter tuning

ML Jargon

Features

Loss function

Training data

Architecture

Architecture

GPU

Batch processing

Generalisation

Mask

Optimisation

Inference

Parameters

Regularisation

Test data

Cross validation

Target variable

Inductive bias

Overfitting

Hyperparameters

Validation set

Bias-variance tradeoff